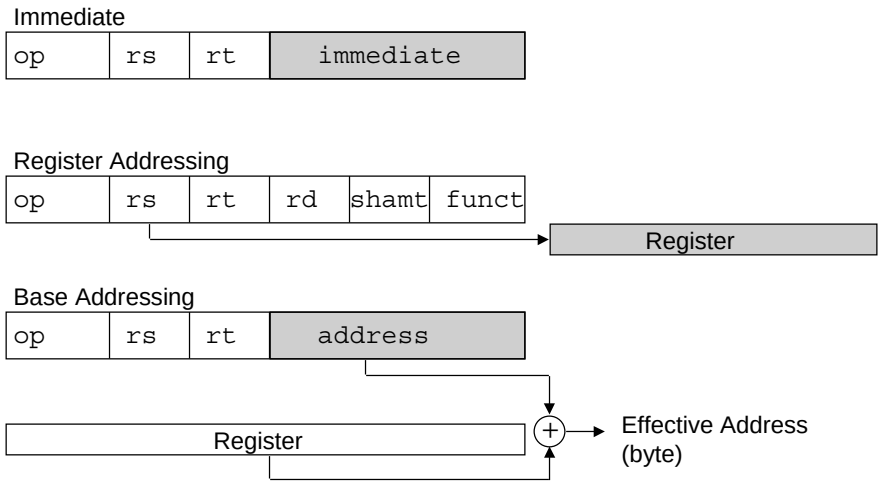
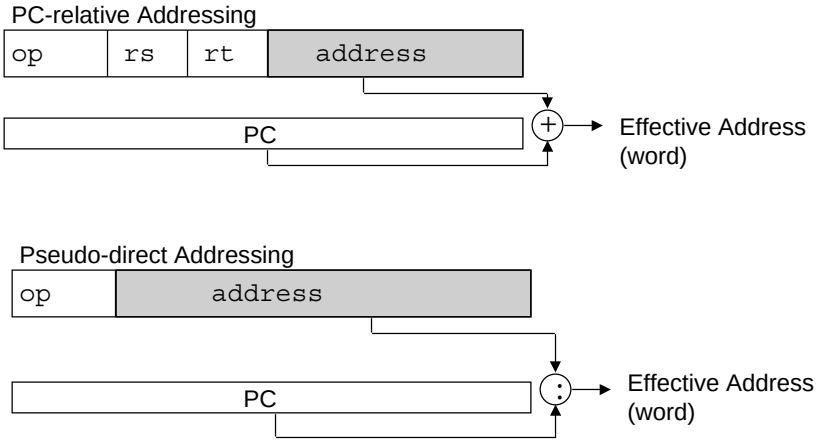
Assembly Language -- More Features

Assembly language is the medium for directly programming the ISA. It reveals the beauty and the quirks of the computer's design

Addressing Modes



Addressing Modes (Continued)



Signed Arithmetic

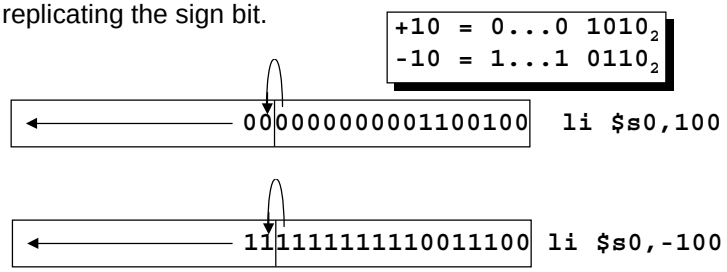
- Most operations involving integers are signed
- When a number is loaded into a larger field, it is sign extended, which preserves its sign

```
                                  0000 0011    3 as byte
                      0000 0000 0000 0011      3 as hw
0000 0000 0000 0000 0000 0000 0000 0011        3 as word

                                  1111 1101    -3 as byte
                      1111 1111 1111 1101      -3 as hw
1111 1111 1111 1111 1111 1111 1111 1101        -3 as word
```

What's Your Sign?

- The standard `add`, `sub`, `addi`, etc. instructions use signed 2s complement values
  - The immediate field is *sign-extended* to make a 32 bit value.
  - A 2s complement value represented with *x* bits can be converted to the same value in a *y* bit representation *y* > *x*, by replicating the sign bit.



Sign extend is also applied when loading bytes and halfwords, i.e. lb and lh

Unsigned Instructions

MIPS provides a set of operations to perform unsigned arithmetic:

```
addu   $8, $9, $10 # add unsigned
subu   $8, $9, $10 # find difference
addiu  $8, $9,  10 # add unsigned immediate
multu  $8, $9, $10 # multiply unsigned
divu   $8, $9, $10 # divide unsigned
lbu    $8, 0($9)   # load byte unsigned
lhu    $8, 0($9)   # load halfword unsigened
```

- Notice that the "u" modifier follows the opcode

The most common application of unsigned arithmetic is for address calculations

ASCII

- American Standard Code for Information Interchange -- now known redundantly as “US-ASCII”
- A 7-bit code for the keyboard characters and certain “control characters”
- When bytes became 8-bits, the coding became extended or 8-bit ASCII

8-bit ASCII

ASCII	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
0 0 0 0	NU	SH	SX	EX	ET	EQ	AK	BL	BS	HT	LF	VT	FF	CR	SO	SI
0 0 0 1	DL	D1	D2	D3	D4	NK	SY	EB	CN	EM	SB	EC	FS	GS	RS	US
0 0 1 0		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
0 0 1 1	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0 1 0 0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0 1 0 1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0 1 1 0	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0 1 1 1	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DT
1 0 0 0	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	SA	SB	SC	SD	SE	SF
1 0 0 1	DC	P1	P2	SE	CC	MW	SP	EP	SS	OS	PM	AC	UC	ES	EH	AP
1 0 1 0	A0	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	-	®	¯
1 0 1 1	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
1 1 0 0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1 1 0 1	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1 1 1 0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1 1 1 1	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Program Forms

- C Program